

## BEST AVAILABLE COPY

(19)



JAPANESE PATENT OFFICE

## PATENT ABSTRACTS OF JAPAN

(11) Publication number: 10011347 A

(43) Date of publication of application: 16.01.98

(51) Int. Cl.

G06F 12/00

G06F 13/00

G06F 13/00

(21) Application number: 08181182

(22) Date of filing: 21.06.96

(71) Applicant: NEC CORP

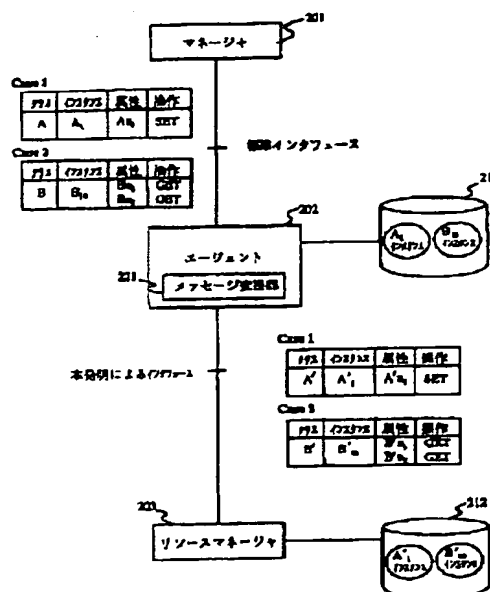
(72) Inventor: KAWAGOE TERUYUKI  
TANAKA KATSUYUKI(54) HARDWARE RESOURCE MANAGEMENT  
MODULE MAKING COMMON SYSTEM

COPYRIGHT: (C)1998,JPO

(57) Abstract:

**PROBLEM TO BE SOLVED:** To reduce the development load of a protocol converting program in an agent and to improve productivity by converting a standard operation which is stipulated by means of a standard interface into an out-of-standard interface between the agent and a resource manager.

**SOLUTION:** The agent 202 which receives an operation request 'SET' judges that the data accompanies the updating of data of the resource manager 203 and transmits the operation request to a message converting part 221 inside the agent 202. The message converting part 221 which receives the request analyzes a message in correspondence with a standard protocol, executes conversion into the form of the message corresponding to the interface between the agent and the resource manager and permits the correspondence of the instant of an object to be managed designated as the object of the operation request by the manager 201 with management object instance in the resource manager 203 to be 1:1.



(19) 日本国特許庁 (JP)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 10 - 1 1 3 4 7

(43) 公開日 平成10年(1998)1月16日

(51) Int. Cl. <sup>6</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F	12/00	5 4 5	G 0 6 F	12/00 5 4 5 A
	13/00	3 5 1		13/00 3 5 1 B
		3 5 3		3 5 3 C

審査請求 有 請求項の数 4

F D

(全 1 0 頁)

(21) 出願番号 特願平8-181182

(22) 出願日 平成8年(1996)6月21日

(71) 出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72) 発明者 川越 照行

東京都港区芝五丁目7番1号 日本電気株式会社内

(72) 発明者 田中 克行

東京都港区芝五丁目7番1号 日本電気株式会社内

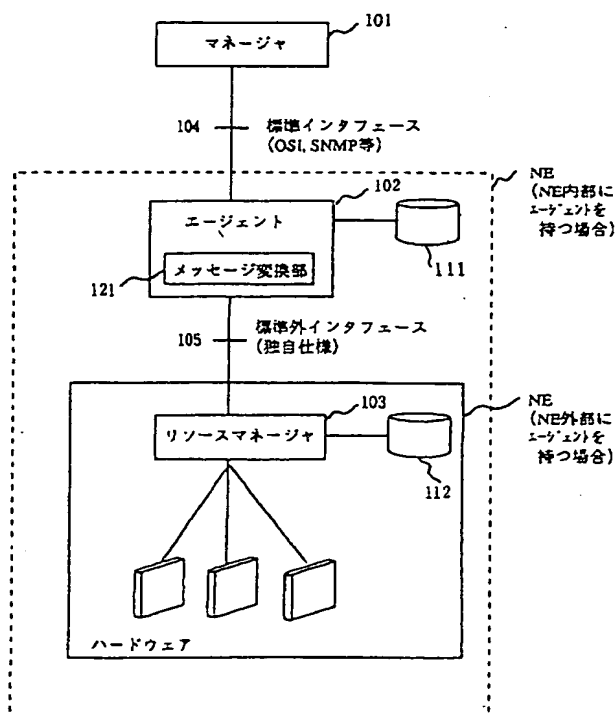
(74) 代理人 弁理士 加藤 朝道

(54) 【発明の名称】 ハードウェアリソース管理モジュール共通化方式

(57) 【要約】

【課題】 O S I 管理、または S N M P 管理を行う標準のマネージャ、エージェントシステムにおいてエージェントとハードウェアのリソース管理を行うリソースマネージャとの間のインタフェースを標準インタフェースの考えに基づき汎用的、且つプロトコルの変換処理を最小限にする方式の提供。

【解決手段】 リソースマネージャにて管理を行うデータを、標準のインタフェースにおける管理情報ベース (M I B) に実装された管理対象 (M O) と同じオブジェクト指向に対応させることにより、M I B とリソースマネージャのインスタンス同士の対応を容易にし、更に汎用的なメッセージのフォーマットを規定することにより、標準プロトコルと標準外のプロトコルとの変換処理を最小限にする。



## 【特許請求の範囲】

【請求項1】OSI (Open Systems Interconnection) またはSNMP (Simple Network Management Protocol) などの標準プロトコルにより管理を行う通信システムにおいて、

管理を行う側（「マネージャ」という）と、管理が行われる対象（「エージェント」という）と、の間の標準インタフェースにて規定されている標準操作を、前記エージェントと、ハードウェアのリソース管理を行うハードウェアリソース管理マネージャ（「リソースマネージャ」という）との間の標準外インタフェースに変換して適用可能としたことを特徴とするハードウェアリソース管理モジュール共通化方式。

【請求項2】前記リソースマネージャが、前記エージェントにおける管理情報ベース (Managed Information Base) におけるオブジェクト指向による管理対象の定義 (Managed Object) と同じ単位でオブジェクトのインスタンスを管理することにより、前記エージェントからの操作要求をハードウェアの操作に置き換える、ことを特徴とする請求項1記載のハードウェアリソース管理モジュール共通化方式。

【請求項3】前記エージェントと、前記リソースマネージャとの間のインタフェースにおけるメッセージのフォーマットが、

オブジェクトヘッダ部と、オブジェクト情報部と、要求パラメータ部と、を備えて構成され、

前記オブジェクトヘッダ部は、メッセージの実行結果を含み、

前記オブジェクト情報部は、前記リソースマネージャにおける管理対象インスタンスの識別子を含み、

前記要求パラメータ部は、属性のタイプと、該属性の識別子、および該属性の値の領域を含む、ことを特徴とする請求項1又は2記載のハードウェアリソース管理モジュール共通化方式。

【請求項4】ハードウェアのリソース管理を行うリソースマネージャが管理するデータの形態を、マネージャとエージェント間の標準インタフェースにおける管理情報ベース (MIB) に実装される管理対象 (MO) と同等のオブジェクト指向に対応させ、前記標準インタフェースに従うオブジェクト指向型操作要求が、前記エージェントと前記リソースマネージャとの間のインタフェースにおいてオブジェクト指向型メッセージに変換されることを特徴とするハードウェアリソース管理モジュール共通化方式。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明はネットワーク管理システムにおけるハードウェアリソース管理方式に関し、

特にOSI (Open Systems Interconnection) 又はSNMP (Simple Network Management Protocol) 等の標準のプロトコルにより装置の管理を行うネットワーク管理システムの管理方式に関する。

## 【0002】

【従来の技術】この種の標準のプロトコルにより装置の管理を行うネットワーク管理システムの従来技術について、図6を参照して説明する。標準プロトコルにより管理を行う通信システムの従来技術においては、管理をされる側であるエージェント602と、ハードウェアのリソースの管理を行うリソースマネージャ603と、のインタフェースは、装置に固有の独自のコマンドによるプロトコルが実装されている。

【0003】また、リソースマネージャ603を構成するプログラムは、オブジェクト指向ではなく専ら機能指向型で作成されていることが多く、マネージャ601とエージェント602間におけるオブジェクト指向により定義された操作要求と、エージェント602とリソースマネージャ603との間における機能指向によるコマンドと、の間では基本的な考え方が相違するため、これらの間のプロトコルの変換を行う装置621が必要とされていた。マネージャ601からエージェント602に対してOSI又はSNMP等の標準インタフェースを介して送信されたオブジェクト指向の操作要求は、プロトコル変換装置621で装置固有の機能指向のコマンド形態にプロトコルに変換され（図示の如く、コマンドと可変長レコード形式の単純なレコード構成）、リソースマネージャ603に装置固有のインタフェースを介して送出される。なお、図6において、611はエージェントのMIBを示し、例えばSNMP等においては、ネットワーク管理情報の構造とそのデータベースも標準化されており、「MIB」と呼ばれている。

【0004】なお、文献（1995年電子情報通信学会、通信ソサイエティ大会、B-448、第115頁）に記載されているようなローカル管理インタフェース（エージェントとリソースマネージャ間のインタフェース）をGDMO (Guideline Definition Managed Object) 形式で定義する方法もあるが、これは独自仕様の定義を表現する手段としてGDMOを用いているだけでローカル管理インタフェースでやり取りされるメッセージの考え方や、その構造、そしてリソースマネージャにて管理される管理対象の定義まで行うものではない。

【0005】すなわち、上記文献「標準管理インタフェースと同様なAPI (Application Programming Interface) でローカル管理インタフェースのアプリケーションを構築することができる」と記載されているが、これよりAPIとして、「GET」や「SET」等の標準の操作を用いることは

容易に類推できるが、これらの操作を行う対象が、どういう単位で、どのように定義されるのか、という点に関しては上記文献では言及されていない。

#### 【0006】

【発明が解決しようとする課題】上記した従来技術は下記記載の問題点を有している。

【0007】第1の問題点は、エージェントとリソースマネージャとの間のインタフェースが装置毎に異なった独自のコマンドによるものであるため、エージェントもしくはリソースマネージャのいずれかの仕様が明確に決定されていないと、もう一方の詳細な仕様が決まらないという、ことである。

【0008】また、第2の問題点は、エージェントとリソースマネージャのソフトウェアの考え方がオブジェクト指向と機能指向とで異なるため、エージェント側またはリソースマネージャ側でオブジェクト指向によるメッセージを機能指向に変換する変換プログラム装置を装置毎に開発しなければならないという、ことである。

【0009】さらに、第3の問題点は、エージェントとリソースマネージャとの間のインタフェースは装置毎の独自仕様であるため、異なった装置に対する管理システムを開発しようとした場合には、エージェントおよびリソースマネージャのインタフェース部分を再度開発し直さなければならないという、ことである。

【0010】従って、本発明は、上記問題点に鑑みてなされたものであって、その目的は、エージェントとリソースマネージャとの間のインタフェースにてやり取りされるメッセージを、OSIやSNMPなどで標準化されている操作要求の形態と容易に対応付けが行えるメッセージとすることにより、エージェントにおけるプロトコル変換プログラムの開発負荷を軽くし、生産性を向上させる、ようにした方式を提供することにある。

【0011】また、本発明の他の目的は、標準のMO（管理対象の定義、Managed Object）で扱うメッセージの形態を取り入れたことにより、異なる装置であっても標準の規格に従った部分はインタフェースを共通にすることを可能とし、異なる装置間でのエージェントまたはリソースマネージャの流用度が高まり、ソフトウェア資源の再利用性を向上させると共に、標準のMOに従っている部分におけるインタフェースを共通化することにより保守性を向上させる方式を提供することにある。

#### 【0012】

【課題を解決するための手段】前記目的を達成するため、本発明のハードウェアリソース管理モジュール共通化方式は、OSI（Open Systems Interconnection）またはSNMP（Simple Network Management Protocol）などの標準プロトコルにより管理を行う通信システムにおいて、管理を行う側（「マネージャ」とい

う）と、管理が行われる対象（「エージェント」という）と、の間の標準インタフェースにて規定されている標準操作を、前記エージェントと、ハードウェアのリソース管理を行うハードウェアリソース管理マネージャ（「リソースマネージャ」という）との間の標準外インタフェースに変換して適用可能としたことを特徴とする。

【0013】本発明においては、前記リソースマネージャが、前記エージェントにおける管理情報ベース（Managed Information Base）におけるオブジェクト指向による管理対象の定義（Managed Object）と同じ単位でオブジェクトのインスタンスを管理することにより、前記エージェントからの操作要求をハードウェアの操作に置き換える、ことを特徴とする。

【0014】また、本発明においては、前記エージェントと、前記リソースマネージャとの間のインタフェースにおけるメッセージのフォーマットが、オブジェクトヘッダ部と、オブジェクト情報部と、要求パラメータ部と、を備えて構成され、前記オブジェクトヘッダ部は、メッセージの実行結果を含み、前記オブジェクト情報部は、前記リソースマネージャにおける管理対象インスタンスの識別子を含み、前記要求パラメータ部は、属性のタイプと、該属性の識別子、および該属性の値の領域を含む、ことを特徴とする。

【0015】本発明の概要を以下に説明する。本発明においては、リソースマネージャにて管理する管理対象をエージェントの管理情報ベース（Managed Information Base、以下「MIB」という）（図1の111）にて管理を行うオブジェクト指向にて定義された管理対象と同じ単位で持つ（図1の112）。

【0016】また、マネージャ（図1の101）からの標準プロトコルに従って操作要求をエージェントのMIB（図1の111）に反映させる場合、その対象となるデータがリソースマネージャ側のデータベース（図1の112）にもある場合、エージェントはエージェント内のメッセージ変換部（図1の121）にて、エージェントとリソースマネージャ間のインタフェースのプロトコルに従いメッセージの変換を行う。

【0017】メッセージ変換部（図1の121）においては、エージェントからリソースマネージャへのメッセージの形態を共通形式（図3の301）に従ったフォーマットに変換し、それぞれの領域に値を設定する機能を持つ。

【0018】本発明のハードウェアリソース管理モジュール共通化方式によれば、リソースマネージャにて管理する管理対象をエージェントのMIB（図1の111）にて管理を行うオブジェクト指向にて定義される管理対象と同じ単位で持つ（図1の112）ようにしたので、

10

20

30

40

50

エージェントのメッセージ変換部（図1の121）では、マネージャ（図1の101）より操作要求の対象として指定されたMOのインスタンスと、リソースマネージャにおける管理対象インスタンスとが、1対1で対応付けられる。

【0019】また、操作の対象となる属性の項目もエージェントにおけるMIB（図1の111）とリソースマネージャにおける管理対象の属性と1対1に対応がとれているため、エージェント内のメッセージ変換部（図1の121）において、容易に標準プロトコルから独自プロトコルへの変換が可能になる。

【0020】

【発明の実施の形態】本発明の実施の形態について、図面を参照して以下に詳細に説明する。図1は、本発明の実施の形態に係るシステムの全体構成を示すブロック図である。

【0021】図1を参照すると、本発明の実施の形態において、システムの全体構成は、管理を行う側の標準プロトコルに基づいたマネージャ101と、管理をされる側のエージェント102との間のメッセージの protocols は、標準にて規定されているOSIのCMIP（Common Management Information Protocol）やSNMP（Simple Network Management Protocol）にて行われ、エージェント102側には管理対象となるインスタンスがMIB111の情報として格納されている。

【0022】マネージャ101がある操作を要求した際に、その操作の対象となるデータの値がエージェント102のMIB111のみに格納されている場合には、エージェント102はMIB111内に格納されている管理対象インスタンスに対する操作のみを行うが、その操作要求がハードウェアに直接関係のある場合には、エージェント102はその内部のメッセージ変換部121に対して操作要求を行う。

【0023】エージェント102から操作要求を受けたメッセージ変換部121は、その要求の詳細を標準インタフェース104におけるメッセージのフォーマットから標準外インタフェース（独自仕様）105への変換を行う。

【0024】リソースマネージャ103が要求を受け付けると、リソースマネージャ103は、その管理情報が格納されているデータベース112の情報に対する処理を行い、必要があれば、所定のハードウェアの設定の変更を行う。なお、システムの構成として管理対象となる装置（NE：Network Element）の内部に、エージェントが実装される形態（図1の破線で囲んだ範囲）と、NEの外部にエージェントが実装される形態（図1の実線で囲んだ範囲）とがある。

【0025】次に、本発明の実施の形態の動作につい

て、図2、図3および図4を参照して説明する。

【0026】まず、初めにマネージャ201は、標準プロトコルに従いエージェント202のMIB211に格納されている管理対象インスタンスに対して操作要求を発行する（図4のステップS11）。

【0027】操作要求を受け取ったエージェント202は、その操作対象となる属性の実体の存在の判断により、エージェント202のMIB211に格納されている管理対象インスタンスのみの変更または値の読出しで良いのか、或いはリソースマネージャの管理しているデータ211の変更または値の読出しが必要であるか否かの判定を行う（図4のステップS12）。

【0028】このとき、その操作要求がエージェント202のMIB211のみの変更または値の読出しでよい場合には、エージェント202はMIB211のデータの変更のみを行い、その結果をマネージャ201へ応答し、処理を終了する（図4のステップS13）。

【0029】一方、マネージャ201からの操作要求がリソースマネージャ203にて管理されるインスタンスに対する変更または値の読出しが必要な場合には、エージェント202はその内部のメッセージ変換部221に対して、その要求を送出する。

【0030】要求を受けたメッセージ変換部221は、エージェント202から受け取ったメッセージの解析を行う（図4のステップS14）。

【0031】この時、メッセージ変換部221は、エージェント202より受け取ったメッセージの中から、

- ①「どのクラスに対する要求か」、
- ②「そのクラスのどのインスタンスに対する要求か」、
- ③「そのインスタンスで持っているどの属性等に対する何の操作か」、及び、
- ④「その属性等に対して具体的に何を行うのか」を解析する。

【0032】次に、メッセージ変換部221は、標準のプロトコルに従ったメッセージを独自のメッセージ形式への変換を行う（図4のステップS15）。このとき、メッセージ変換部221では、図3の301に示すように、オブジェクトヘッダ部、オブジェクト情報部、及び要求パラメータ部からなる共通形式を作り出す。

【0033】より具体的には、オブジェクトヘッダ部は、全メッセージ長や応答の場合に成功、失敗等の結果が入る実行結果のフィールド等を具備する。

【0034】また、オブジェクト情報部は、エージェントより受け取った「どのクラスに対する要求か」、「そのクラスのどのインスタンスに対する要求か」を解析した結果をそれぞれ図3の312におけるクラス識別子および区分1～nに設定する。

【0035】ここで区分1～nは、リソースマネージャ203が、そのクラスのインスタンスを一意に識別するための階層化された識別番号である。

【0036】要求パラメータ部は、要求された操作がいくつかの属性に対応するのかを表す属性情報数とその属性の詳細な情報を表す属性情報部（図3の313）からなり、さらに属性情報部は、図3の314として示すように、属性タイプ、属性識別子、そして属性値のフィールドからなり（図3の314）、属性タイプのフィールドは、より具体的には、図3に315として示すように、属性の基本的な型の種別を表す属性情報種別と、その具体的な型を表す属性型コードと、からなる。

【0037】属性識別子のフィールドは、指定されたインスタンスの持っている属性のうち、どの属性なのかを識別するための識別子を持ち、また属性値のフィールドはその指定された属性の値が設定される。

【0038】上記したメッセージのフォーマットにメッセージ変換部221が、エージェント202からのメッセージを変換した後、当該変換したメッセージをリソースマネージャ203に対して送出する。

【0039】リソースマネージャ203で要求を受け付けると（図4のステップS16）、操作要求をリソースマネージャ203の管理対象であるインスタンス212に対して実行する（図4のステップS17）。

【0040】このとき、リソースマネージャ203にて管理を行うインスタンスに対する処理が成功したか、失敗したかをリソースマネージャ203は判断し（図4のステップS18）、失敗した場合には、その応答を前記の共通形式にしたがった形でメッセージ変換部221を介してエージェント202へ報告し、その処理を終了する（図4のステップS19）。

【0041】一方、成功した場合には、リソースマネージャ203はその応答を同じくメッセージ変換部221を介してエージェント202へ報告する。成功の応答を受け取ったエージェント202は、エージェントのMIB211に操作要求の結果を反映する（図4のステップS20）。なお、操作要求が値の読出しの場合はエージェントのMIB211のデータの更新は行わない。

【0042】エージェントMIB211の情報の更新が終わると、エージェント202はマネージャに対して処理の成功を通知し（図4のステップS21）、その処理を終了する（図4のステップS22）。

#### 【0043】

【実施例】上記した本発明の実施の形態をより具体的な例を以て説明すべく、本発明の第1の実施例について図面を参照して説明する。

【0044】図2を参照すると、マネージャ201からエージェント部202に対してOS1のCMIPにて規定された標準のMOであるクラスAのインスタンスであるA<sub>1</sub>の持つ属性A<sub>11</sub>に対する操作要求「SET」が要求されたとする（標準プロトコルにおけるCase1参照）。

【0045】なお、条件として、このクラスの値は直接

ハードウェアの設定に関係があるため、リソースマネージャ203にて管理されるデータであり、そのデータの型は単純型の2バイトの整数値（short型）とする。

【0046】操作要求「SET」を受けたエージェント202は、そのデータがリソースマネージャ203のデータの更新を伴うことを判断し、エージェント202内のメッセージ変換部221に対して、この操作要求を送信する。

【0047】要求を受け取ったメッセージ変換部221は、標準のプロトコルに従ったメッセージを解析し、本発明（請求項3）によるインタフェースに対応したメッセージの形式に変換を行う。

【0048】ここでの具体的な変換としては、図5を参照すると、まずオブジェクトヘッダ部511のフォーマットを作成し、その中に要求元を識別するための識別子（この場合、エージェント202を識別するためのID）と、要求先を識別するための識別子（この場合、リソースマネージャ203を識別するためのID）を設定する。

【0049】また、オブジェクトヘッダ部511においては、この他にも、例えばある要求とその応答との対応を識別するためのインボーク識別子や、そのメッセージが要求であるのか、応答であるのか、またはリソースマネージャからの通知であるのか、というメッセージの種別を表すメッセージ種別のフィールド等が設けられ、メッセージ変換部221は、それらの値も設定する。

【0050】次に、メッセージ変換部221は、オブジェクト情報の設定をエージェント202から指定されたクラスと、インスタンスの情報と、に基づき、メッセージの変換を行う。

【0051】この例の場合、クラスAのインスタンスA<sub>1</sub>が、例えばハードウェアのシェルフ1の中のスロット5に実装されているカードの機能8に対応するものであれば、オブジェクト情報部（図5の512）に設定される値として、クラス識別子は、クラスAに対応するリソースマネージャにて管理を行うクラスA'を表す“0001”、階層数は“3”、そして各階層の区分を表すフィールドには、前記シェルフ1、スロット5、機能8を表すように区分1には“1”、区分2には“5”、そして区分3には“8”が設定される。

【0052】次に、メッセージ変換部221は、要求パラメータ部（図5の513）の設定を行うが、この例の場合、SET要求された属性A<sub>11</sub>に対応するリソースマネージャ203にて管理を行う属性A'の識別子は“1001”であるため、属性識別子のフィールド（図5の514）には“1001”が設定され、また属性値のフィールドには、SET要求にて更新される値が設定される。

【0053】また、属性タイプのフィールド（図5の5

15)には、属性情報種別としては、単純型を表す“1”が、また属性型コードには2バイトの整数(short)を表す番号“1”が設定される。

【0054】なお、この例の場合、エージェント202からSETの要求された属性は一つであるので要求パラメータ部(図5の513)における属性情報数は“1”であり、属性情報1~nも一つのフィールドのみ使用することになる。

【0055】このようにして、上記情報が設定し終わると、メッセージ変換部221は、最後に、全メッセージ長の値を、オブジェクトヘッダ部(図5の511)のメッセージ長のフィールドに書き込み、メッセージの解析および変換の処理を終了する。

【0056】変換されたメッセージは、メッセージ変換部221より、リソースマネージャ203へ送信され、リソースマネージャ203では送信されてきたメッセージに従い管理を行っている管理対象インスタンスのデータ(図2の212)であるクラスA'のインスタンスA'の属性A'に対して値の更新の処理を行う。

【0057】処理が正常に終了した場合、リソースマネージャはその応答を、本発明(請求項3)によるインタフェースのメッセージの形式にて、エージェント202に対して送信し、エージェント202はOSI標準にて規定されているCMIPのメッセージ形式にて、マネージャ201へその応答を送信し、その処理を終了する。

【0058】次に、本発明の第2の実施例について同じく図2を参照して説明する。

【0059】マネージャ201からエージェント202に対して、OSIのCMIPにて規定された標準のMOであるクラスBのインスタンスであるB<sub>10</sub>の持つ属性B<sub>11</sub>およびB<sub>12</sub>に対する操作要求「GET」が要求されたとする(図2の標準プロトコルにおけるCase2参照)。なお、条件として、このクラスの値は直接ハードウェアの設定に関係があるためリソースマネージャ203にて管理されるデータであり、そのデータの型は属性B<sub>11</sub>が2バイトの整数値(short型)のリスト型であり、属性B<sub>12</sub>が文字列(string型)の配列型とする。

【0060】操作要求「GET」を受けたエージェントは、そのデータがリソースマネージャ203のデータの更新を伴うことを判断し、エージェント内のメッセージ変換部221に対して、その要求を送信する。

【0061】要求を受け取ったメッセージ変換部221では、標準のプロトコルに従ったメッセージを解析し、本発明(請求項3)によるインタフェースに対応したメッセージの形式に変換を行う。具体的な変換は、図5を参照すると、まずオブジェクトヘッダ部(図5の511)のフォーマットを作り出し、その中に要求元を識別するための識別子(この例の場合、エージェント202を識別するためのID)と、要求先を識別するための識

別子(この例の場合リソースマネージャ203を識別するためのID)を設定する。

【0062】また、オブジェクトヘッダ部(図5の511)においては、この他にも例えばある要求とその応答との対応を識別するためのインボーク識別子や、そのメッセージが要求であるのか、応答であるのか、またはリソースマネージャ203からの通知であるのかという、メッセージ種別を表すメッセージ種別のフィールド等があり、メッセージ変換部221は、これらの値も設定する。

【0063】次に、メッセージ変換部221は、オブジェクト情報の設定をエージェント202から指定されたクラスとインスタンスの情報に基づきメッセージの変換を行う。この例の場合、クラスBのインスタンスB<sub>10</sub>が例えばハードウェアのシェルフ1の中のスロット3に実装されているカードの機能6の中の5番に対応するものであれば、オブジェクト情報部(図5の512)に設定される値として、クラス識別子はクラスBに対応するリソースマネージャにて管理を行うクラスB'を表す“0002”、階層数は“4”、そして各階層の区分を表すフィールドには前記シェルフ1、スロット3、機能6、番号5を表すように区分1には“1”、区分2には“3”、区分3には“6”、そして区分4には“5”が設定される。

【0064】次に、メッセージ変換部221は要求パラメータ部(図5の513)の設定を行うが、この例の場合、GET要求された属性B<sub>11</sub>とB<sub>12</sub>の2種類であるため属性情報数には2が設定される。

【0065】次に、「属性情報1又は集合属性情報1」の領域においては属性B<sub>11</sub>が対応するリソースマネージャ203にて管理を行う属性B'の識別子は2001であるため属性識別子のフィールド(図5の514)には“2001”が設定され、また「属性情報2又は集合属性情報2」の領域においては属性B<sub>12</sub>が対応するリソースマネージャにて管理を行う属性B'の識別子は“2002”であるため属性識別子のフィールド(図5の514)には“2002”が設定される。

【0066】なお、GETの操作要求の場合、リソースマネージャ203にて管理を行っているデータの値を読み出してくるため、エージェント202からの要求においては属性値(図5の514)のフィールドの値は何も設定されていない。

【0067】次に、「属性情報1又は集合属性情報1」における属性タイプのフィールド(図5の515)に、属性B'に対応した属性情報種別として、リスト型を表す“2”が、また属性型コードには、2バイトの整数(short)を表す番号“1”が設定され、また「属性情報2又は集合属性情報2」における属性タイプのフィールド(図5の515)は、属性B'に対応した属性情報種別としては配列型を表す“3”が、また属性型

コードには文字列 (string) を表す番号 “3” が設定される。

【0068】上記したような情報を設定し終わると、メッセージ変換部 221 は、最後に全メッセージ長の値をオブジェクトヘッダ部 (図 5 の 511) のメッセージ長のフィールドに書き込み、メッセージの解析および変換の処理を終了する。

【0069】変換されたメッセージはメッセージ変換部 211 よりリソースマネージャ 203 へ送信され、リソースマネージャ 203 では、送信されてきたメッセージに従い管理を行っている管理対象インスタンスのデータ (図 2 の 212) であるクラス B' のインスタンス B' <sub>10</sub> の属性 B' <sub>a1</sub> および B' <sub>a2</sub> に対して値の読出しの処理を行う。

【0070】処理が正常に終了した場合、リソースマネージャはその応答を、本発明によるインタフェースのメッセージ形式に、読み出した値を設定するが、この例の場合、属性の型はリスト型と配列型、つまり両方とも集合属性であるため、「属性情報又は集合属性情報」 (図 5 の 514) の属性値のフィールドには、集合属性の場合の属性値 516 で示した集合属性の場合のフォーマットが適用され、要素数のフィールドには読み出された属性の要素の数が設定され、各々の要素の値を表す要素 1 から n のフィールドには、図 5 に 517 で示したような、要素の番号とその読み出された値 (属性 B' <sub>a1</sub> の場合は 2 バイトの整数型、属性 B' <sub>a2</sub> の場合は文字列型) が設定され、エージェントに対して GET 要求に対する応答を送信する。

【0071】応答を受け取ったエージェント 202 は OSI 標準にて規定されている CMIP のメッセージ形式にて、マネージャ 201 へその応答を送信し、その処理を終了する。

【0072】

【発明の効果】以上説明したように、本発明によれば、下記記載の効果を奏する。

【0073】本発明の第 1 の効果は、リソースマネージャにて管理する管理対象をエージェントの MIB にて管理を行うオブジェクト指向にて定義された管理対象と同じ単位で持つため、エージェント内のメッセージ変換部では、マネージャより操作要求の対象として指定された MO (管理対象オブジェクト) のインスタンスと、リソースマネージャにおける管理対象インスタンスと、の対応が 1 対 1 に付けられる。また、本発明によれば、操作の対象となる属性の項目についても、エージェントにおける MIB とリソースマネージャにおける管理対象の属性と、1 対 1 に対応がとれる。このため、メッセージ変換部においては、容易に標準プロトコルから独自プロトコルへの変換が可能になる。

【0074】次に、本発明の第 2 の効果は、エージェント

メッセージのフォーマットを、好ましくは、上記実施の形態 (図 3 参照) で説明したような形式にすることにより、OSI の CMIP 等で用いられる抽象構文である ASN.1 (Abstract Syntax Notation 1) による属性の値の規定を容易にハードウェアへの実際の設定のためのビットマップ形式に変換することができる、ということである。

【0075】本発明の第 3 の効果は、前記のような汎用的なメッセージフォーマットを規定しておくことにより、エージェントとリソースマネージャの管理するオブジェクトの独立性が高まり、システム全体としての仕様が固まっていなくても、それぞれのオブジェクトのスパイラル設定が可能となるという、ことである。

【0076】本発明の第 4 の効果は、前記のような汎用的なメッセージフォーマットを規定しておくことにより異なる装置間でのエージェントとリソースマネージャの流用が可能になることによりソフトウェアの共通化が実現でき、システムのコストを削減できる、ということである。

20 【図面の簡単な説明】

【図 1】本発明を実施の形態に係るシステムの全体構成を示す図である。

【図 2】本発明の一実施例を説明するための図であり、特に標準プロトコルと本発明によるインタフェース部のメッセージ形式の対応を表す図である。

【図 3】本発明の一実施例を説明するための図であり、エージェントとリソースマネージャとのインタフェースにおけるメッセージのフォーマットの概要を表す図である。

30 【図 4】本発明の一実施例の動作を説明するためのフローチャートである。

【図 5】本発明の一実施例におけるエージェントとリソースマネージャとの間のインタフェースにおけるメッセージのフォーマットを表す図である。

【図 6】従来技術の構成を示す図であり、特に標準プロトコルと従来の機能指向による独自インタフェース部のメッセージ形式の対応を表す図である。

【符号の説明】

101 マネージャ (OSI または SNMP 等の標準プロトコルに従ったマネージャ)

102 エージェント (OSI または SNMP 等の標準プロトコルに従ったエージェント)

103 リソースマネージャ

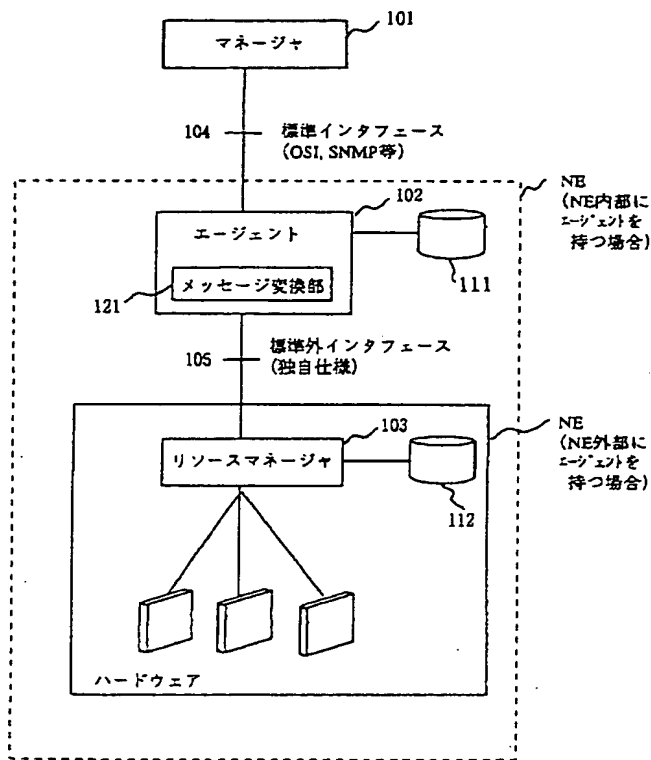
111 MIB (エージェントにて管理を行う標準の管理情報モデルの規定にしたがった管理対象インスタンスを格納する MIB)

112 データベース (リソースマネージャにて管理を行う管理対象インスタンスが格納されているデータベース)

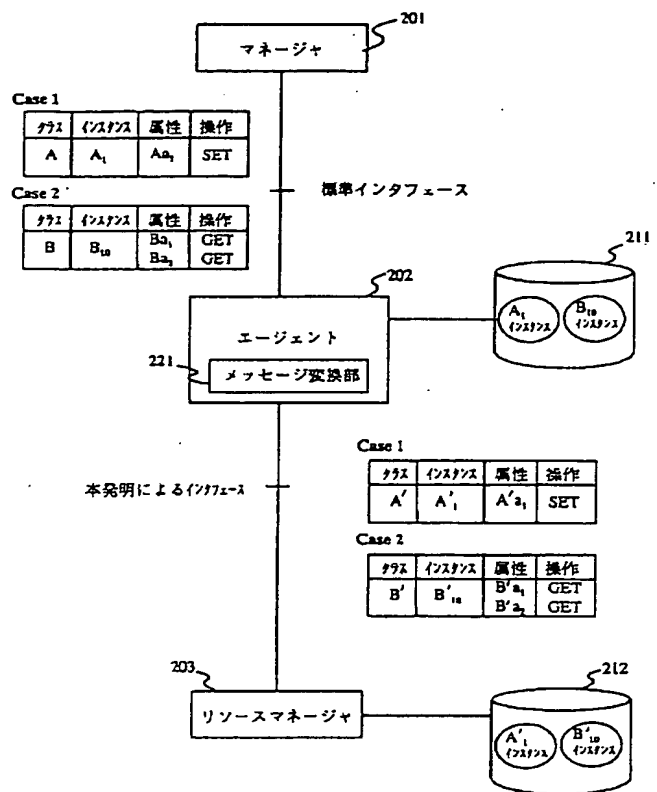
50 121 メッセージ変換部



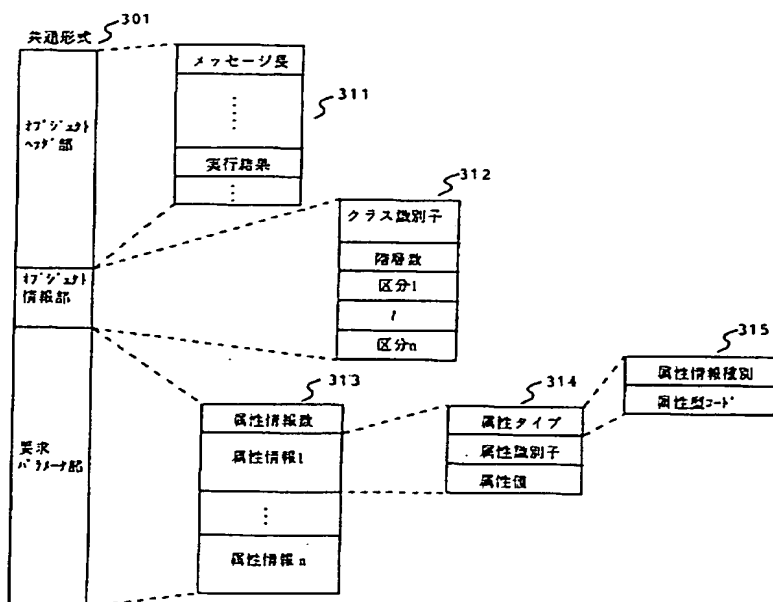
【図1】



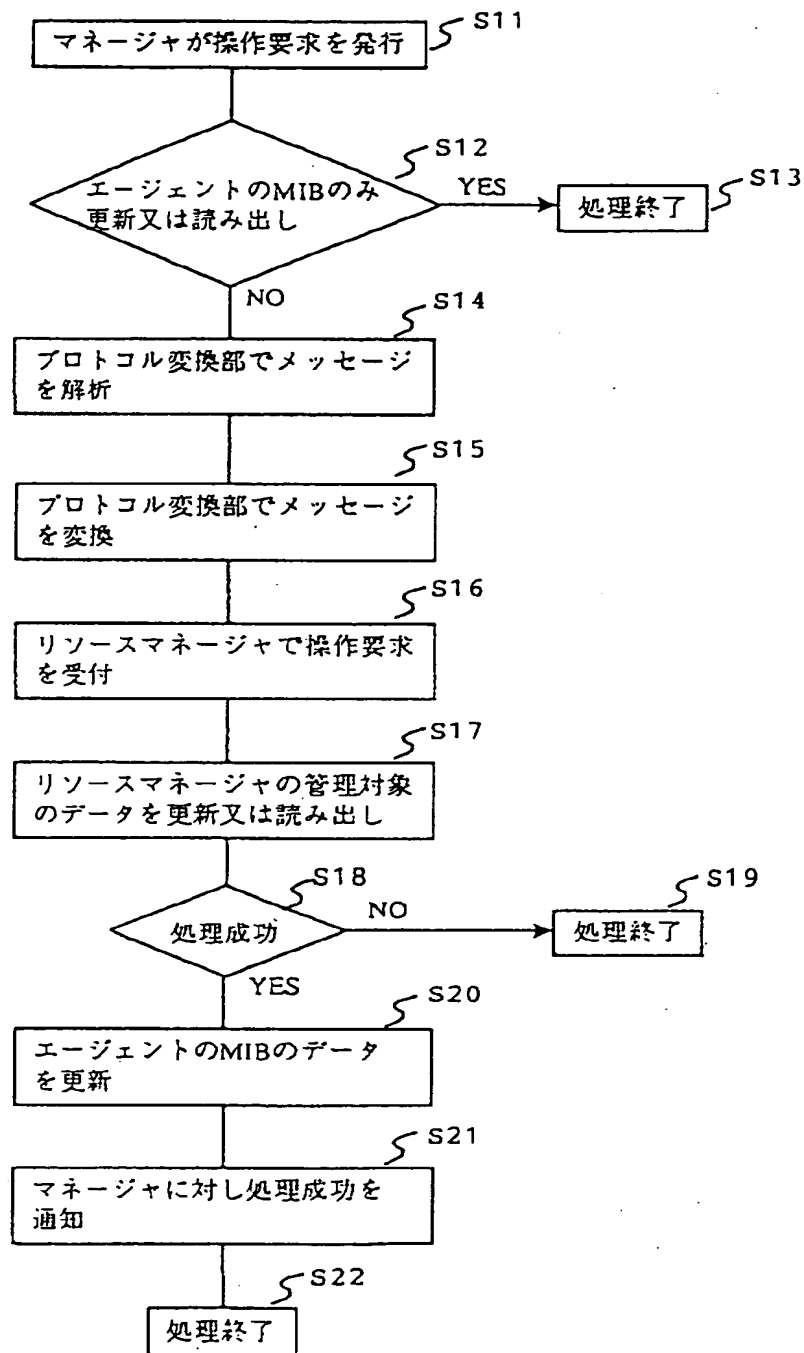
【図2】



【図3】



【図4】



[illegible]

Figure 1 illustrates the system configuration. A central vertical line represents the **標準インタフェース** (Standard Interface). At the top is the **マネージャ** (Manager) 601. Below it is the **エージェント** (Agent) 602. At the bottom is the **リソースマネージャ** (Resource Manager) 603. To the right of the Agent is a database 611 containing **A<sub>1</sub> (インスタンス)** and **B<sub>1</sub> (インスタンス)**. To the right of the Resource Manager is a database 612 containing **GET\_A'a'→**, **GET\_B'a'→**, and **GET\_B'a'→**. On the left, **Case 1** and **Case 2** are shown as tables. Case 1 table: | ナム | インスタンス | 属性 | 操作 |; | A | A<sub>1</sub> | Aa<sub>1</sub> | GET |. Case 2 table: | ナム | インスタンス | 属性 | 操作 |; | B | B<sub>1</sub> | Ba<sub>1</sub>, Ba<sub>2</sub> | GET |. A label **621** points to the **プロトコル変換装置** (Protocol Conversion Device) located between the Agent and the Resource Manager. A label **従来のインタフェースの例** (Example of the conventional interface) points to the central vertical line.

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☒ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**